

WESTERN SYDNEY
UNIVERSITY



Module 4

Overview of Logical Operators

Logical Operators

Operator

&

|

~

xor()

Operation

Logical **AND**

Logical **OR**

Logical **NOT**

Logical **exclusive-OR**

Logical Operators

- Operators that are applied to logical values to produce a logical result.
- For the purpose of logical operations, “numeric zero” is treated as a logical zero.
- Logical operations can be applied to numbers of non-logical data type (int, double, etc., i.e. normal numbers). These numbers are classified as “non-zero”

Truth Table: Logical NOT (\sim)

X	Y	$\sim Y$
F	F	T
F	T	F
T	F	T
T	T	F

T = True, F = False

- The logical NOT operator (\sim) returns
 - TRUE when the operand is FALSE
 - FALSE when the operand is TRUE
- For non-logical numbers:
 - FALSE when the operand is non-zero
 - TRUE when the operand is zero

Truth Table: Logical AND (&)

X	Y	X & Y
F	F	F
F	T	F
T	F	F
T	T	T

T = True, F = False

- The logical AND operator (&) returns
 - TRUE when both operands are TRUE
 - Otherwise FALSE
- For non-logical numbers:
 - TRUE when both operands are non-zero
 - Otherwise FALSE

Truth Table: Logical OR (|)

X	Y	X Y
F	F	F
F	T	T
T	F	T
T	T	T

T = True, F = False

- The logical OR operator (|) returns
 - TRUE when either operand is TRUE
 - Otherwise FALSE i.e. both are FALSE
- For non-logical numbers:
 - TRUE when either operand is non-zero
 - Otherwise FALSE i.e. both are zero

Truth Table: Logical exclusive-OR

X	Y	xor(X,Y)
F	F	F
F	T	T
T	F	T
T	T	F

T = True, F = False

- The logical XOR operator (exclusive-OR, xor()) returns
 - TRUE when either operand is TRUE
 - FALSE when both operands are TRUE
 - FALSE when both operands are FALSE
- For non-logical numbers:
 - TRUE when either operand is non-zero
 - FALSE when both operands are non-zero
 - FALSE when both operands are zero

Logical Operators: Examples

Input:

```
a = true;
```

```
b = false;
```

```
c = true;
```

Can you guess the results of:

```
a&b           0 (False)
```

```
b&c           0
```

```
a&c           1 (True)
```

```
a|b           1
```

```
b|c           1
```

```
a|c           1
```

```
c|a           1
```

```
b|b           0
```